第 37 卷第 5 期
录用定稿
计算机应用研究
Application Research of Computers
Vol. 37 No. 5
Accepted Paper

非数值化特征的条件概率区域划分(CZT)编码方法 *

贺　亮[1]，徐正国[1]，李　赟[1]，沈　超[2]

(1. 盲信号处理重点实验室，成都 610041; 2. 西安交通大学 智能网络与网络安全教育部重点实验室，西安 710049)

摘　要：非数值化特征经常出现在数据中，对其有效编码是采用机器学习模型解决问题的关键。针对目前被广泛使用的 one-hot 编码方法的编码结果具有较大的稀疏性，并且编码出的数值仍然没有明确的物理意义等问题，提出一种基于条件概率的区域划分编码算法 CZT(conditional-probability-based zone transformation coding)。该方法首先对特征进行条件概率计算，并依据条件概率划分特征区域，按照区域内的联合条件概率进行编码；然后将 CZT 编码算法与 one-hot 算法进行对比分析，从理论上推导并证明 CZT 编码对特征的压缩率至少为每个特征取值空间的平均大小，同时证明经过 CZT 编码后的问题具有更简单的优化目标形式，利于设计后续机器学习算法；最后通过采用相同结构的神经网络进行分类，在 titanic 数据集下对比 CZT 算法和 one-hot 算法编码数据后对分类器性能的影响，结果表明 CZT 编码的数据在分类准确率和稳定性均有提升。

关键词：深度学习；非数值化特征；特征工程；联合条件概率编码

中图分类号：TP391　　doi: 10.19734/j.issn.1001-3695.2018.10.0818

Conditional-probability zone transformation coding for categorical features

He Liang[1], Xu Zhengguo[1], Li Yun[1], Shen Chao[2]

(1. *National Key Laboratory of Science & Technology on Blind Signal Processing*, *Chengdu* 610041, *China*; 2. *MOE Key Laboratory for Intelligent Networks & Network Security*, *Xi'an Jiaotong University*, *Xi'an* 710049, *China*)

Abstract: Categorical features always exist in the dataset and coding them is a key issue for solving problems efficiently by machine learning models. One-hot coding is a wide accepted method to convert the features into feature values, and however it attracts sparse space and meaningless value after coding. To improve the coding performance, a novel coding method based on conditional probability after dividing the features into zones, which is called CZT coding (Conditional-probability-based Zone Transformation coding) , is designed. The CZT coding calculates the conditional probability of each feature and then divides the features into several zones and finally coding the features in each zone. This paper mathematically proved that compared with the state-of-the-art method - one-hot coding, CZT coding reduces the code length by at least the mean of feature spaces and the issue switches into an easier one after CZT coding for the following machine learning model. Finally, using the same neuron network as the classifier, the performance of CZT coding and one-hot coding is compared by using the titanic dataset, and the result is that CZT coding makes the classifier performs better both on the accuracy and steadiness.

Key words: deep learning; categorical features; feature engineering; conditional probability

0　引言

随着数据存储技术的发展，大量数据被存储，人工智能技术得以飞速发展。传统的机器学习算法中，模型的参数较少，少量数据即可对参数进行估计，对大数据量利用率较低，在算法模型选定后，刻画数据分布的函数形式随之限定，只需要根据提供的数据寻找合理的函数参数。而深度学习技术中，由链接结构的千变万化导致模型参数可以迅速增长，增加了神经网络的学习能力，使得深度学习在数据较大时仍然可以有效学习到数据的特征并且提高网络的性能。传统机器学习算法和深度学习算法在不同数据量下的性能对比[1]示意图如图 1 所示。

传统的机器学习技术中，算法的参数较少，从而限制了算法在大量数据下对数据的利用能力。以支持向量机(support vector machine, SVM)为例，在考虑一般的核函数的情况下，SVM 为 Lagrange 对偶乘子法，相应的分类平面即为数据维度空间下的线性模型，模型参数只与数据维度数量相当，即使数据再多，也无法拓展参数形式，分类器只取决于少量的支持向量，无法在数据增大的过程中对数据进行利用以挖掘更多数据特征。SVM 的优势是可以在数据量较少的情况下，给出有效的支持向量用于分类。

神经网络由边连接相应节点构成，其 VC 维(vapnik-chervonenkis dimension)[2]是节点数和边数的乘积，在有效的训练学习算法前提下，该网络可以逼近任何几乎处处连续的函数。但是由于 VC 维高，训练时需要的数据一般下认为是 10 倍的 VC 维，大数据量条件下，更适用于使用深度神经网对数据进行训练和学习以刻画和描述数据特征。在数据增多的情况下，深度学习仍然能有效学习出数据特征。在很多实际应用中表明，采用 SVM 进行分类器设计的算法，如果改成深度神经网络，则相应的算法速度、性能等方面在

收稿日期：2018-10-15；**修回日期：**2018-12-03　**基金项目：**国家自然科学基金重点项目（U1736205）；国家自然科学基金项目（61773310）

作者简介：贺亮（1990-），男，黑龙江佳木斯人，工程师，博士，主要研究方向为人工智能、网络协议分析（lianghe@sei.xjtu.edu.cn）；徐正国（1986-），男，工程师，博士，主要研究方向为智能信息处理；李赟（1984-），男，工程师，博士，主要研究方向为网络态势感知；沈超（1984-），男，副教授，博士，主要研究方向为智能信息处理、智能穿戴设备行为分析.

大数据条件下都将有所提升[3~5]。可见，对于目前数据量逐渐增大的趋势下，深度学习能够更好地进行拓展和使用。

深度学习以其强大的函数拟合能力和学习能力促进了人工智能和机器学习领域的发展，数据量的增长又反作用于深度学习，使其能够更加有效地学习数据内在的关系，从而分析出数据的潜在价值。实际问题中，数据中的特征经常是非数值化的，如性别、颜色、语言、文字[6]等，而神经网络需要处理数值化的输入，对这一类非数值化特征的处理，有一种比较常见的做法是采用 one-hot 编码或是词嵌入方法[7~9]。然而这一类方法会引入较大的向量空间冗余，并且具体的编码数值没有明确的数值意义。虽然深度学习用较深的神经网络来解决特征工程的问题[10]，期望通过位于前端的几层神经元对数据自动进行预处理，以达到对数据进行特征变化、特征提取等特征工程的问题，而实际中训练该特征工程网络层需要大量的训练数据和较高的调参数技巧，往往造成深度神经网络训练时间过长，甚至数据量不够而导致无法得到好的训练效果[11]。实际数据集中常遇到取值广泛的非数值化属性的情况，例如，话音中进行通信的用户双方在通话网络规模较大时用户数量也将会比较庞大，并且每个用户都是非数值化的取值；在自然语言处理中，每个单词便是一个非数值化取值，在文字量巨大的语言中，单词这一属性的取值空间将非常庞大；网络协议层中的 IP 地址也是非数值化的，为了利用 IP 地址这一属性，也需要对 IP 地址进行数值化编码；某些具有庞大用户群体的手机应用中，用户 ID 较多且是非数值化，为了挖掘用户的规律习惯等信息，需要对 ID 进行数值化便于后续机器学习算法进行分析等场景。如果采用 one-hot 编码对上述列举的情况进行编码，则会导致编码结果是一个较为稀疏的高维向量，并且每一个向量中只用一位为 1 表示特定的用户、单词或者 IP 地址等。本文针对上面一类应用场景中的非数值化特征，提出一种基于区域划分的条件概率编码方法——CZT 算法，以解决编码空间稀疏以及编码数值无意义的问题。数据量对传统机器学习和深度学习性能的影响如图 1 所示。

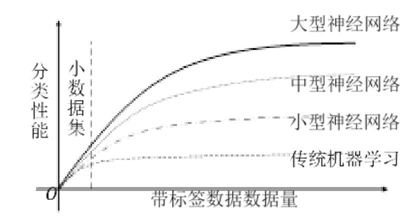


图 1　数据量对传统机器学习和深度学习性能的影响

Fig. 1　Influence of dataset amount on performance of traditional machine learning and deep learning

1　非数值化特征及编码

非数值化特征是数据集中经常遇到的一类属性。例如，在话音网络中，用户 ID 虽然被存储为数字，但是没有明确的数值意义，而对于机器学习算法，需要将这一类非数值化特征进行编码以供后续分类器利用。目前广泛接受的编码方法是 one-hot 编码，即采用高维空间内的具有唯一非零值分量的高维向量作为特征的编码结果。本章介绍非数值化特征常用的预处理方式。

1.1　非数值化特征

在采用机器学习算法进行分类时，默认输入的都是数值化的特征，各个特征通过一定的预处理后参与模型参数的计算，机器学习就是通过不断在数据上进行训练从而不断调优模型参数，以使模型更好地适应数据的过程[12]。数值化的特征一般也未必能直接使用，例如利用身高体重进行机器学习时，身高的单位一般为厘米而体重为千克，不同单位量纲的数据之间对数值直接相加不具有任何意义，因此常用的是对数据进行归一化操作，即对数据 X 做如下处理：

$$\hat{X} = \frac{X - \mathbb{E}X}{\sqrt{\mathbb{D}X}},$$

在经过减去均值除以标准差的操作后，采用 $\hat{X}$ 作为输入训练数据，可以避免量纲的影响，各个特征之间的数值可以进行数学操作作为机器学习的训练数据。然而实际中更多的时候获得的是非数值化的特征，即特征的各种取值之间不具有明确的数值意义。例如，对于颜色这一特征，假设特征的取值空间为 $\{红, 黄, 蓝\}$，则不能简单地认为可以把其分别编码为 $1,2,3$，因为这里的数值是有数学意义的，即数值是有序的，而编码过程指定各个特征的映射是随机的。编码后的实数中，红和蓝分别为 1 和 3，其差值比红和黄大，这些特征很可能被之后的机器学习模型所利用，而实际上该特征并没有这一特征，是在编码过程中人为引入了这一额外的数值特性。为了使编码不具有偏序性，需要考虑采用 one-hot 编码对这一类非数值化的特征进行编码。

1.2　one-hot 编码

one-hot 编码又被译为独热码或一位有效码，也缩写成 OHC 编码，其编码过程是根据特征的取值空间，设计相应的编码向量长度，并将相应的特征取值位置设置为 1，其余为 0 。one-hot 编码实现如下从特征到编码空间的映射，如果某一特征的取值空间是 $\mathcal{S}$ ，则对于保序的特征空间 $\hat{\mathcal{S}}$ ，其第 i 个元素 s_i 的编码结果为 $c \triangleq (c_1,\ldots,c_{|\mathcal{S}|}), ?c_j = 1_{j=i}, j = 1,2,\ldots,|\mathcal{S}|$ 。任何时候，one-hot 编码的结果中只有一位有效位并且取值为 1，其余位取值均为 0 。对非数值化数据先进行编码，得到的数值输入到分类器进行训练和分类，该方法的流程一般如图 2 所示。

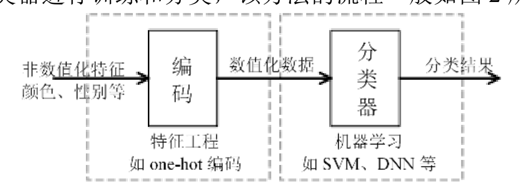


图 2　特征工程与机器学习的架构关系

Fig. 2　Frame of feature engineering and machine learning

2　条件概率区域编码算法——CZT

针对 one-hot 编码中编码结果维度高，只有唯一有效值而使输入数据转换成稀疏数据并且有效位数值没有物理意义等问题，本文提出基于条件概率密度的区域编码算法——CZT 算法。该方法首先对非数值化特征进行区域划分，划分的原则是条件概率密度具有相同的取值空间；然后针对划分的区域结果对同一区域内的各个特征按照条件联合概率进行编码。

2.1　非数值化特征的数学描述

对于某一个具体的分类问题，假设问题具有 n 个标签 $l_1,l_2,\ldots,?l_n$ ，这里的◦表示元素具有顺序，不妨设重点关注的标签为 l_1 ，各个标签对应的样本数据个数分别为 $m_1,m_2,\ldots,m_n$ ，则样本总量为

$$m \triangleq \sum_{i=1}^{n} m_i,$$

数据集特征有 $特征1,特征2,\ldots,特征K$，相应的数据集结构如表 1 所示。为了简化表述，所考虑数据均为非数值化特征，各维特征的非数值化取值空间分别为 $\mathcal{S}_1,\mathcal{S}_2,\ldots,\mathcal{S}_K$，并且每个取值空间具有元素个数分别为 $s_1,s_2,\ldots,s_K$， 即 $s_i \triangleq |\mathcal{S}_i|, i=1,2,\ldots,K$， 可记 $\mathcal{S}_i=\{F_1,F_2,\cdots,F_{s_i}\}, i=1,2,\cdots,K$， 即实际数据中的属性 i 的取值为 $f_{i,\circ}\in S_i$。虽然特征的取值是非数值化的，仍然可以用实数刻画，不妨 $F_\circ\in\mathbb{R}$。例如，投掷立方体的取值空间是立方体的六个平面，但仍然可以将其标记为实数，只是这时的具体数值没有实数意义，即不能表明标记为1的平面与标记为6的平面有任何数值上1与6的关系，对这样的实验求期望也是没有意义的，然而深度学习方法较强的学习能力可能会学习到这一类关系。另一方面，还可以认为 $F_\circ$ 是特征事件集合到实数 $\mathbb{R}$ 的泛函，即对相应特征取值的实数映射。对于特征 i，取值为 $f_{i,j}=F_j\in\mathcal{S}_i$ 记为事件 ω_j，该特征的全部事件记为 Ω_i， 可以构造随机变量 $X_i(\omega_j)=F_j$， 即对每一维特征 i， 都可以看成是一个随机变量 $X_i:\Omega_i\to S_i\subset\mathbb{R}$，本文直接用 X_i 表示特征 i。对于非数值化特征的问题，随机变量是离散的，下面讨论的问题均以概率形式出现，涉及概率密度概念时,如无特殊说明,指该随机变量取值空间稠密但可以无一致连续要求的情况下,近似的概率密度曲线,此时概率密度为函数微元。

表 1　非数值化数据集结构

Table 1　Structure of dataset with categorical features.

特征1	特征2	特征3	⋯	特征K	标签
$f_{1,1}$	$f_{2,1}$	$f_{3,1}$	⋯	$f_{k,1}$	l_1
$f_{1,2}$	$f_{2,2}$	$f_{3,2}$	⋯	$f_{k,2}$	l_1
⋯	⋯	⋯	⋯	⋯	⋯
f_{1,m_1}	f_{2,m_1}	f_{3,m_1}	⋯	f_{k,m_1}	l_1
f_{1,m_1+1}	f_{2,m_1+1}	f_{3,m_1+1}	⋯	f_{k,m_1+1}	l_2
⋯	⋯	⋯	⋯	⋯	⋯
f_{1,m_1+m_2}	f_{2,m_1+m_2}	f_{3,m_1+m_2}	⋯	f_{k,m_1+m_2}	l_2
⋯	⋯	⋯	⋯	⋯	⋯
$f_{1,m-m_n}$	$f_{2,m-m_n}$	$f_{3,m-m_n}$	⋯	$f_{k,m-m_n}$	l_n
⋯	⋯	⋯	⋯	⋯	⋯
$f_{1,m}$	$f_{2,m}$	$f_{3,m}$	⋯	$f_{k,m}$	l_n

每个特征的概率分布，其取值空间是离散的实数，记 $\mathcal{V}(X_i)$ 表示特征 i 的取值空间。定义属性 i 和 j 属于同一个**区域**(zone)，如果对于两个特征具有相同的取值空间，即 $\mathcal{V}(X_i)=\mathcal{V}(X_j)$。被划分到同一个区域内的各个特征的联合概率分布，称为这个区域的概率分布。本文提出的 CZT 算法依据实际数据集的特点，找到使划分出的各个区域具有不同的概率分布取值空间作为编码依据。

2.2　算法原理

首先考虑对单个特征进行编码的情况，在关注标签 l_1 情况下，各个特征的概率函数为

$$P_{X_i}(x|l_1)\triangleq\Pr\{X_i=x\,|\,l_1\},$$

将特征 X_i 相应的取值 x 编码为 $P_{X_i}(x|l_1)$。

对于标签 l_1 下的两个特征 X_i,X_j， 若 $P_{X_i}(x|l_1)=P_{X_j}(x|l_1)$， 此时如果将两个特征编码成相同的数字，容易造成混淆，因此需要采用 $P_{X_i,X_j}(x_i,x_j|l_1)$ 联合分布作为两个特征的编码。如果联合分布仍然与某个特征 X_k 具有相同的编码结果，则继续采用 $P_{X_i,X_j,X_k}(x_i,x_j,x_k|l_1)$ 作为编码，直到不存在编码重复的结果为止。这里具有相同的编码结果的单个特征，在考虑联合分布后整体作为编码的若干特征，是在特征列表中按照一个区域考虑的，因此被称为划分到同一个区域中进行联合编码，如图 3 所示。本算法也是基于该划分区域的思想进行编码的，因此命名为 CZT 算法。据此法进行编码，一方面，可以增加对非数值化特征的编码能力，降低编码结果的维度；另一方面，可以降低稀疏性。

图中将在标签 l_1 下具有相同条件概率分布取值空间的特征划分为同一个区域，并用 $Z_1,Z_2,\ldots,Z_k$ 表示各个区域，对于同一个区域内的属性，采用联合条件概率密度进行编码，从而降低编码后数据的维度，并使编码数值具有条件概率的物理意义。

对于不同的标签 $\{l_1,l_2,\ldots,l_n\}$， 存在 n_1,n_2， 使得 $P_{X_i}(x|l_{n_1})=P_{X_i}(x|l_{n_2})$ 时，如果采用特征编码，则两个标签下无法区分各个样本实际的不同。此时，也需要考虑结合其他特征共同编码，即进行区域划分。然而与上一种情况不同，下面分析如何选择参与共同编码的特征。

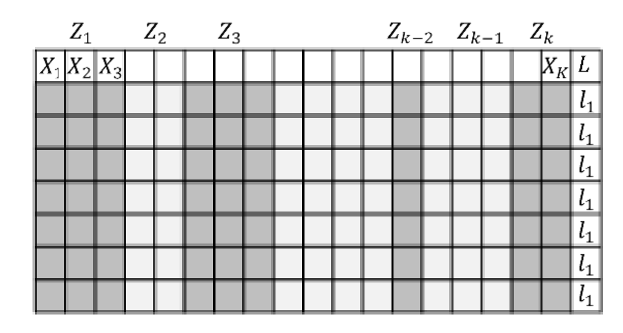


图 3　CZT 算法区域(zone)示意图

Fig. 3　Schematic diagram of zone in CZT algorithm.

首先需要引入函数距离的定义：对连续型随机变量的概率密度函数 $f_1(x),f_2(x)$， 其距离按照如下 KL 散度(K-L divergence)：

$$\mathcal{L}(f_1(x),f_2(x))\triangleq\int_{-\infty}^{+\infty}f_1(x)\log\frac{f_1(x)}{f_2(x)}\mathrm{d}x$$

对离散型随机变量的概率分布 $p_1(x),p_2(x)$,其距离定义为

$$\mathcal{L}(p_1(x),p_2(x))\triangleq\sum_x p_1(x)\log\frac{p_1(x)}{p_2(x)}$$

基于如上定义，选择 X_j 满足

$$X_j=\operatorname*{argmin}_X\mathcal{L}(P_{X_i}(x|l_{n_1},l_{n_2}),P_X(x|l_{n_1},l_{n_2})),$$

即与重复编码一致的概率分布最近的特征划分到同一区域进行联合编码。当特征较多时，该方法需要遍历所有特征以找到与当前特征概率分布最接近的一个，可以引入一个距离阈值 θ，当某个 X_j 与 X_i 的概率分布之距离小于该阈值时便可以选定该特征作为划分到同一区域的对象，进行联合编码，即

$$\exists X_j \in \{X|\mathcal{L}(P_{X_i}(x|l_{n_1},l_{n_2}),P_X(x|l_{n_1},l_{n_2}))<\theta\},$$

将 X_j 与 X_i 划分到同一个区域。如果 θ 选择不合理，如选取较小，上面的集合可能为空，但经过各个距离函数的遍历计算，已经可以找到距离最小的 X_j 作为区域划分和联合概率编码的特征。

若找到满足上述条件的 X_j 后，联合编码在 l_{n_1},l_{n_2} 下仍然相同，即 $P_{X_i,X_j}(x_i,x_j|l_{n_1})=P_{X_i,X_j}(x_i,x_j|l_{n_2})$，则依上法继续寻找 X_k，直到编码不同。下证明，如果将全体特征均选为同一个区域作为联合分布编码，编码结果针对标签 l_{n_1},l_{n_2} 相同，则此时的标签 l_{n_1},l_{n_2} 不可分。

当所有 K 个特征在标签 l_{n_1},l_{n_2} 下属于同一个区域，即联合分布相同时，记 $\vec{\mathcal{X}}\triangleq(X_1,X_2,\ldots,X_K)^{\mathrm{T}}$，相应随机向量的取值记为 $\vec{x}\triangleq(x_1,x_2,\ldots,x_K)^{\mathrm{T}}$ 为同一个区域，即

$$P_{\vec{\mathcal{X}}}(\vec{x}|l_{n_1})=P_{\vec{\mathcal{X}}}(\vec{x}|l_{n_2}),$$

给定样本下，该样本属于这两个标签的概率之比值为

$$\frac{P_{\vec{\mathcal{X}}}(l_{n_1}|\vec{x})}{P_{\vec{\mathcal{X}}}(l_{n_2}|\vec{x})}=\frac{P_{\vec{\mathcal{X}}}(\vec{x}|l_{n_1})\Pr\{l_{n_1}\}}{P_{\vec{\mathcal{X}}}(\vec{x}|l_{n_2})\Pr\{l_{n_2}\}}=\frac{\Pr\{l_{n_1}\}}{\Pr\{l_{n_2}\}},$$

即比值为固定值，而这两个标签下的数据量在训练数据下不变，此时无论样本 $\vec{x}=(x_1,x_2,\cdots,x_K)^{\mathrm{T}}$ 如何选取，都不影响样本条件下两个标签 l_{n_1},l_{n_2} 的概率分布比值，因此无法对这两个标签进行区分，说明数据在这两个标签情况下不可分，对数据进行编码工作也无法区分这两类数据。因此，如果数据是可分的，CZT 算法编码时便不会出现所有特征都划分到同一个区域内的情况。

2.3　CZT 编码算法流程

结合上面的算法原理，给出 CZT 算法流程如算法 1 所示。算法流程中，第一个 for 循环是对单个关注标签下的非数值化特征进行编码，主要考虑该标签下某些特征可能具有相同的编码结果，需要进一步进行区域划分，对其进行联合编码。第二个 for 循环主要针对不同标签下，存在同一特征编码相同的情况，此时需要进一步选择与这个特征的概率分布足够接近的特征划分为同一个区域并联合编码。流程中使用了 for-break-else 语句，其表示的含义是对 for 循环里面的内容，如果执行了 break 语句退出 for 循环，则不执行 else 语句内的操作，如果 for 循环成功遍历了所有操作并且没有执行 break 语句，则跳出循环后执行 else 内的操作。算法的核心功能主要是在划分数据集某些特征为若干区域后，对区域内的特征依据该区域的条件概率分布进行编码，编码结果使得各个区域具有互不相同的取值空间。

算法 1　CZT 编码算法流程

输入：　非数值化特征 $X_1,X_2,\ldots,X_K$ 的 m 个样本；
联合编码阈值 $\theta>0$。
输出：　特征的数值化编码矩阵。

for　X_i　in　$[X_1,X_2,\ldots,X_K]$：
　计算 $P_{X_i}(x|l_1)$ 作为 X_i 的编码
　for　X_j　in　$[X_1,X_2,\ldots,X_i]$：
　　if　$P_{X_i}(x|l_1)=P_{X_j}(x|l_1)$：
　　　将 (X_i,X_j) 划分到一个区域
　　　计算 $P_{X_i,X_j}(x_i,x_j|l_1)$ 作为 (X_i,X_j) 的编码
　　　if 存在 X_k 的编码相同：
　　　　继续递归划分区域并编码
for　X_i　in　$[X_1,X_2,\ldots,X_K]$：
　for　l_s　in　$[l_1,l_2,\ldots,l_n]$：
　　计算 $P_{X_i}(X_i|l_s)$
　　for　l_t　in　$[l_1,l_2,\ldots,l_s]$：
　　　if　$P_{X_i}(x|l_s)=P_{X_i}(x|l_t)$：
　　　　#寻找恰当的特征所属的区域
　　　　for　X_j　in　$[X_1,X_2,\ldots,X_K]$：
　　　　　计算存储 $\mathcal{L}(P_{X_i}(x|l_s,l_t),P_{X_j}(x|l_s,l_t))$
　　　　　if　$\mathcal{L}(P_{X_i}(x|l_s,l_t),P_{X_j}(x|l_s,l_t))<\theta$：
　　　　　　将 (X_i,X_j) 划分到一个区域
　　　　　　将 $P_{X_i,X_j}(x_i,x_j|l_s,l_t)$ 作 (X_i,X_j) 编码
　　　　　　break
　　　　else:
　　　　　求 $\underset{X}{\operatorname{argmin}}\,\mathcal{L}(P_{X_i}(x|l_s,l_t),P_X(x|l_s,l_t))$
　　　　　将 (X_i,X_j) 划分到一个区域
　　　　　将 $P_{X_i,X_j}(x_i,x_j|l_s,l_t)$ 作为 (X_i,X_j) 编码
　　　　if 存在编码相同:
　　　　　继续递归寻找区域并编码

2.4　CZT 编码算法复杂度分析

对于计算不同标签下的联合概率编码时，引入 θ 可以在一定误差范围内提前跳出联合概率计算的循环，然而最差情况下需要完全遍历所有已经计算的特征，因此针对不同标签，当发生编码重复时，计算复杂度为 $O(mK^2)$，这里的 $O(\circ)$ 表示

高阶无穷大渐近项。对于 m 个样本，需要统计 K 个特征的概率分布，基本操作的复杂度即为 $\mathrm{O}(mK)$，综合两者分析结果，CZT 编码算法时间复杂度为 $\mathrm{O}(m^2K^3)$，为多项式时间。

3　CZT 编码算法性能分析

CZT 算法对特征进行编码后，编码的空间不会具有很高维度，使得后续机器学习需要处理的问题得到简化。本章分别从 CZT 算法对特征空间编码后与 one-hot 相比的压缩率、后续机器学习算法待解决的优化问题以及算法的准确率等方面对 CZT 算法的性能给出理论推导。

3.1　特征空间压缩率

在 one-hot 编码中，每种非数值化特征的编码长度是由该特征的取值空间大小决定的，即特征 i 的编码长度为 s_i，并且只有一个值为 1，其余为 0，即 $(c_1,c_2,\ldots,c_{s_i}),c_j\in\{0,1\},\sum_{j=1}^{s_i}c_j=1$，并且 c_j 的数值大小没有具体物理意义，只是代号数值化。从而对于每一个数据，其各个特征的编码结果为

$$s\triangleq\sum_{i=1}^{K}s_i,$$

输入矩阵将是一个 $m\times s$ 的矩阵，每一行的行和为 K。如果实际特征取值较为广泛，特征取值空间大，该矩阵将会是一个很稀疏的矩阵。

采用 CZT 编码时，如果每一维特征都具有能区分的编码，即划分出的各个区域只包含一个特征，不需要联合概率编码，此时每一维特征的非数值化取值都用相应的条件概率编码，因此只需要 $m\times K$ 的输入矩阵。若存在需要联合概率编码的情况，以及划分出的区域中存在多个特征，则只会比当前的输入维度更小，即 $m\times k(k<K)$ 的矩阵，矩阵中的数据都是非零数值，数据代表着各个特征或者联合特征的统计概率，具有一定物理意义，方便后续分类器利用该数值。

对比 CZT 编码和 one-hot 编码，可以看出 CZT 编码的改进如下：a)编码出的矩阵维度大大降低；b)编码出的矩阵数据由稀疏矩阵变为非稀疏矩阵；c)矩阵的元素数值有具体的含义，不再是符号的简单数值化表达。CZT 编码算法对特征空间的压缩率为

$$\frac{m\times s}{m\times k}=\frac{s}{k}\geq\frac{\sum_i^K s_i}{K}\triangleq\bar{s},$$

压缩率至少为 $\bar{s}$，即经过 CZT 编码算法后，每个数据的编码结果压缩率至少为每个特征取值空间的平均大小。

3.2　编码数据维度降低对分类问题的简化

经过 one-hot 编码的数据点稀疏分布在高维空间内，即分布在 s 维空间的晶格内，这里的晶格即为边长取 1 的高维立方体，并且严格有 K 维取值为 1，而经过 CZT 编码的向量，各个维度含义是区域的条件概率，相应分布在最高 K 维空间内，并且在空间内各个维度取值在 $(0,1]$ 之内取值。

在原始系数空间内，晶格这一条件便是数据的内在联系，而这一联系对于分类器分类意义不大，对应之前提过的 one-hot 编码的 0 和 1 没有具体数值意义。对原始进行 CZT 编码后的空间分布更具有条件概率的实际意义，并且取值几乎处处连续。原始数据经过 one-hot 编码后的数据为 $C\in\mathbb{R}^s$，且其中的任意一个元素 $c_i\in\{0,1\}$，$\sum_{i=1}^{s}c_i=K$，经过 CZT 编码后的数据为 $S\in\mathbb{R}^k\subset\mathbb{R}^K$。对于 one-hot 编码，没有对数据进行加工，可以看做是对非数值化数据的直接语义编码，即 CZT 编码算法也可以处理 one-hot 编码的结果，对其进行区域划分并统计条件概率作为编码结果，因此存在 $\mathcal{F}\in\mathbb{R}^{s\times K}$，$S=\mathcal{F}(C)$。对于一个分类问题，可以看成在一定数据条件下最优化一定指标的问题，即

$$\min_D L(D,C;\mathcal{F})。$$

这里的优化变量即为分类器需要学习的分类平面，用 D 表示，数据集也可以是 CZT 算法编码过的数据集 S。经过 CZT 编码后的数据取值空间可以认为是 $(0,1]$ 上几乎处处连续的空间，而 one-hot 编码则是离散的 $0-1$ 取值空间。采用 CZT 算法编码数据使后续机器学习算法待解决的问题的复杂度相应得到降低。

3.3　编码结果在空间分布的稳定性

对于数据集 X，采用 one-hot 编码得到的编码结果中，每一维特征 X_i 编码成一个长度为 s_i 的一维向量，并且该向量只有一个元素为 1，其余为 0，因此方差为

$$\mathbb{D}X_i=\mathbb{E}X_i^2-\mathbb{E}^2X_i=\frac{1}{s_i}-\frac{1}{s_i^2}=\frac{s_i-1}{s_i^2}\sim\frac{1}{s_i}。$$

如果采用 CZT 编码，编码的结果至多为 K 维，并且向量的每一位都是条件概率，介于 0~1 间，根据 CZT 编码规则，相应的条件概率和为 1，即 $\sum_{i=1}^{K}x_i=1$。相应的方差为

$$\mathbb{D}X_i=\mathbb{E}X_i^2-\mathbb{E}^2X_i=\frac{K\sum_{i=1}^{K}x_i^2-\left(\sum_{i=1}^{K}x_i\right)^2}{K^2}=\frac{K\sum_{i=1}^{K}x_i^2-1}{K^2}。$$

下面对该结果进行方差分析，首先需要引入如下引理，对于一系列随机变量 $X_1,X_2,\ldots,X_n$，$X_i\in[0,1]$，且 $\sum X_i=1$，当 $X_n=1$ 且 $X_1=X_2=\cdots=X_{n-1}=0$ 时方差最大，当各个 X_i 均取值为均值 $\bar{X}_i$ 时，方差最小。如果每一个随机变量不相等，且最小相差 ε 时，相应的取法类似，即令前 $n-1$ 个随机变量分别从 0 以 ε 为间距取值，方差最大；令所有随机变量以 ε 为间隔取值在均值 $\bar{X}_i$ 左右时，方差最小。

极限状态下，如果某个编码的条件概率接近 1，其他接近 0 时，上面的方差项最大，相应上确界为

$$\sup\mathbb{D}X_i=\frac{K-1}{K^2},$$

在 CZT 编码中，保证各个编码结果互不相等，相应条件为 $|x_i-x_j|>\varepsilon$，此时相应的上确界可以认为在有 $K-1$ 个编码取值分别为 $0,\varepsilon,2\varepsilon,\ldots,(K-2)\varepsilon$，剩余一个编码结果为 $1-\frac{1}{2}(K-1)(K-2)\varepsilon$ 时取到，相应的方差上确界

$$\sup\mathbb{D}X_i=\mathrm{O}(K^3)\varepsilon^2-\frac{1}{K}+\frac{K-1}{K^2},$$

方差取值最小，对应各个编码结果在均值附近取值。编码结果的均值为 $\bar{X}_i=\frac{1}{K}$，则令各个 X_i 在均值附近以 ε 为间隔取值。在 K 为偶数时，$K\triangleq2K_0$，各个 X_i 取值为

$$\frac{1}{K}-(K_0-1)\varepsilon-\frac{\varepsilon}{2},\frac{1}{K}-(K_0-2)\varepsilon-\frac{\varepsilon}{2},\ldots,\frac{1}{K}-\frac{\varepsilon}{2},$$

$$\frac{1}{K}+\frac{\varepsilon}{2},\ldots,\frac{1}{K}+(K_0-1)\varepsilon+\frac{\varepsilon}{2},$$

K 为奇数时，$K \triangleq 2K_0+1$，各个 X_i 取值为

$$\frac{1}{K}-K_0\varepsilon,\ldots,\frac{1}{K}-\varepsilon,\frac{1}{K},\frac{1}{K}+\varepsilon,\ldots,\frac{1}{K}+K_0\varepsilon,$$

相应方差的下确界为

$$\inf \mathbb{D}X_i=\begin{cases}\dfrac{K^2+4}{8}\varepsilon^2 & \text{偶数维特征}\\ \dfrac{K^2-1}{12}\varepsilon^2 & \text{奇数维特征}\end{cases},$$

对于下确界，CZT 编码方法相应量级在 $\frac{K^2}{10}\varepsilon^2$，而 one-hot 编码为 $\frac{1}{s_i}$，而由于 $s_i \gg K$，从而 $\varepsilon=\frac{\sqrt{10}}{K\sqrt{s_i}}\sim\frac{1}{K^{\frac{3}{2}}}$ 时两算法下确界相当。在 CZT 编码过程中，可以控制相差较大程度后的条件概率密度作为编码结果，因此 ε 在实际操作时可以取到合理的数值，至少保证 $\varepsilon \gg \frac{\sqrt{10}}{K\sqrt{s_i}}$，此时相应的 CZT 编码的方差结果较 one-hot 编码结果的数据分布形式方差更大，即类间离散度大，利于后面的分类器对数据进行分类，后面的实验可以验证，经过 CZT 编码后的数据用于分类时准确率和稳定性均有提升。

4 CZT 编码效果对比

为了对比分析 CZT 算法的性能，对 titanic 数据集分别在 CZT 算法和 one-hot 编码下，使用相同的神经网络结构对 Titanic 生还人员进行预测，然后针对实验结果对比分析本文提出的 CZT 编码算法的性能优势。

4.1 数据集简介

本文采用 titanic 数据集[13]作为实验对象，因为该数据集的非数值化特征比较多，并且结构简单，对其分类的研究比较透彻，易于与传统编码方法对比性能。该数据集根据乘客的基本信息，预测 Titanic 遇难时的生还情况。数据集中包括乘客的 ID 号、舱位等级、性别、年龄、上船地点、船舱号、船票价位等信息，其中很多均是非数值化特征。数据集具有部分缺失数据信息，需要采用众数或均值进行补全。使用该数据集对每名顾客的生存概率进行预测，一方面可以看成是对概率进行 logistic 回归，另一方面也是对是否存活这一标签的二分类问题。本文将采用神经网络进行分类，对该数据集输出层只需要一个简单的神经元即可实现相应的回归或是二分类问题。

4.2 分类器设计

为了探究 CZT 算法的性能，尽量降低由分类器自身影响而造成的分类错误情况或对数据的分类能力不足的问题，本文采用具有较强学习能力的神经网络作为分类器。分类器采用五层神经网络，该网络较传统神经网络的三层结构有所加深，但仍不足以称之为深度神经网络，因此此处仅称其为神经网络。同时，经过实验验证，五层神经网络针对该数据集已经具备较好的分类能力。具有五层的神经网络仍然需要反向传播算法修正神经权向量时的梯度弥散的问题，前四层网络的激活函数采用 ReLU 函数[14]，该函数具有如下激活形式

$$\mathrm{ReLU}(x)=\max\{0,x\}$$

其在正半轴导数为 1，负半轴导数为 0，可以避免梯度弥散问题，并已经在深度神经网络中被广泛采用。由于预测的是二值问题，采用 sigmoid 函数作为输出层的激活函数，则对应的输出值直接即为所需的分类结果，网络结构如图 4 所示。

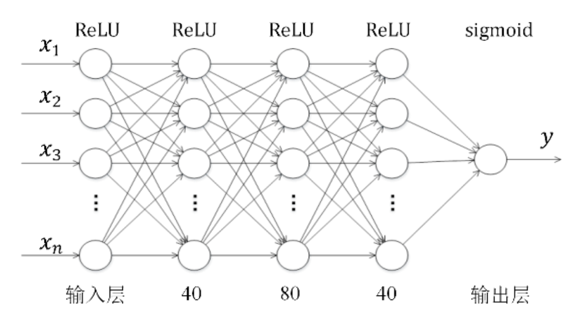


图 4 性能对比时采用的神经网络结构示意图

Fig. 4 Schematic diagram of neural network that is used in performance evaluation.

网络的输入维度即为经过编码后的特征维度。对于采用 one-hot 编码的数据，由于每一个非数值特征的取值范围广泛，会产生稀疏高维的输入向量，而采用 CZT 编码的数据，该输入维度可以显著下降，同时向量中的每个分量的具体数值表示该特征对应的条件概率值，数值之间也具有明确的物理意义，有利于后层神经网络提取相应的特征信息。为了对比 one-hot 编码和 CZT 编码算法性能，采用的神经网络除了输入层神经元个数需要匹配编码后数据的维度，其余层次结构均相同。

4.3 算法性能

分别在原始 titanic 数据集上进行 one-hot 和 CZT 编码，编码出的数据维度分别为 196 维和 8 维，对应的神经网络的输入层也分别是 196 个和 8 个神经元。对数据集进行随机划分，采用 10 折交叉检验(10-fold validation)，重复 10、500 和 1 000 次的实验结果如表 2 所示，表中的 OHC 表示 one-hot 编码。

表 2 CZT 编码算法性能对比

Table 2 Performance evaluation of CZT and one-hot coding(OHC) algorithm.

编码方式	重复次数	网络规模	错误率			
			平均	方差	最小	最大
OHC	10	356	3.670	4.827	2.694	4.938
CZT	10	168	2.402	1.531	1.796	3.143
OHC	500	356	5.084	2.411	2.357	10.44
CZT	500	168	4.400	1.183	0.337	9.764
OHC	1000	356	3.614	1.111	1.684	10.10
CZT	1000	168	2.929	0.892	1.571	9.764

从表中可以看出：a）OHC 算法编码出的数据维度高，导致后端需要采用更加复杂的网络输入层结构对其进行学习；b）CZT 算法错误率下降，错误率方差降低，说明编码出的特征更有利于神经网络训练学习，得到的神经网络的性能更加稳定。图 5 是分别在 10 、 250 、 750 和 次实验的结果用提琴图展示的效果。

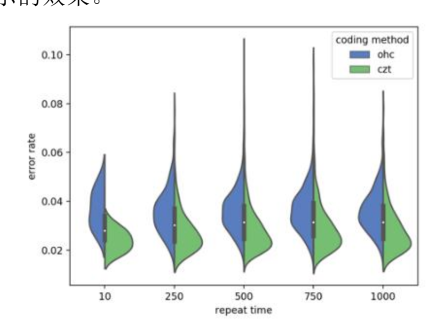


图 5 重复实验不同次数对应错误率的提琴图

Fig. 5 Violin figure of error rates in various experiments

图中横轴是重复的实验次数，纵轴是对应算法的分类错误率，提琴图中每个提琴型左侧蓝色表示采用 one-hot 编码重复实验后整个实验的错误率分布情况，右侧绿色对应 CZT 编码算法。重复实验较少时，如 10～250 次，可以看出 CZT 编码算法的错误率方差较低。当重复到一定情况时，编码方式对算法错误率的影响逐渐稳定，两种编码算法对分类器错误率分布影响基本稳定。可以从错误率的分布提琴图中看出，CZT 算法的错误率较 one-hot 编码对数据处理后分类器错误率下降。

5　结束语

针对 one-hot 等编码方式处理的数据，其结果具有高维度、稀疏性等问题。 CZT 编码算法根据特征的条件概率特点对数据各维特征进行区域划分，并将同一区域的属性共同编码，编码出的数据维度低，同时相应的取值代表该特征区域的条件概率，具有一定的物理意义，为后续的分类器分类提供了较好的数据预处理结果。经过证明，CZT 编码算法能够至少压缩各维特征取值空间大小的平均值倍数的编码长度，并且实验结果表明，CZT 编码算法使分类器分类错误率下降，分类结果的稳定性提升。

参考文献：

[1] Yann L, Yoshua B, Geoffrey H. Deep learning [J]. Nature, 2015, 521 (7553): 436.

[2] Xu Jianhua. Designing nonlinear classifiers through minimizing VC dimension bound [C]// Proc of International Symposium on Neural Networks. Berlin: Springer, 2005: 900-905.

[3] Wang Xiaolong, Shrivastava A, Gupta A. A-Fast-RCNN: hard positive generation via adversary for object detection [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition.2017: 3039-3048.

[4] Chen Xinlei, Gupta A. An implementation of Faster RCNN with study for region sampling[EB/OL]. (2017-02-08) [2018-10-15]. https://arxiv. org/abs/1702. 02138.

[5] Sun Xudong, Wu Pengcheng, Hoi S C H. Face detection using deep learning: an improved faster RCNN approach[EB/OL]. (2017-01-28) [2018-10-15]. https://arxiv. org/abs/1701. 08289.

[6] Lai Siwei, Liu Kang, Xu Liheng, *et al.* How to generate a good word embedding [J]. IEEE Intelligent Systems, 2016, 31 (6): 5-14.

[7] Upadhyay S, Faruqui M, Dyer C, *et al.* Cross-lingual models of word embeddings: an empirical comparison [C]// Proc of the 54th Annual Meeting of the Association for Computational Linguistics. 2016: 1661-1670.

[8] Balaji K, Nikaash P, Raghavender G. Learning vector-space representations of items for recommendations using word embedding models [M]// Procedia Computer Science. 2016: 2205-2210.

[9] Lauren P, Qu G, Yang Jucheng, *et al.* Generating word embeddings from an extreme learning machine for sentiment analysis and sequence labeling tasks [J]. Cognitive Computation, 2018 (3): 1-14.

[10] Ledig C, Theis L, Huszar F, *et al.* Photo-realistic single image super-resolution using a generative adversarial network [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2016: 105-114.

[11] Trishul C, Yutaka S, Johnson A, *et al.* Project adam: building an efficient and scalable deep learning training system [C]// Proc of Usenix Conference on Operating Systems Design and Implementation. [S.l.]: USENIX Association, 2016: 571-582.

[12] Bishop C M. Pattern recognition and machine learning (information science and statistics) [M]. New York: Springer-Verlag, 2006 (4): 499.

[13] Titanic Dataset. Kaggle[EB/OL]. https://www. kaggle. com/c/titanic/.

[14] Xavier G, Antoine B, Bengio Y. Deep sparse rectifier neural networks [C]// Proc of International Conference on Artificial Intelligence and Statistics. 2011: 315-323.